

SYSTEM AND METHOD FOR CLUSTER DEPLOYMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

5 The present application is related to U.S. Patent
Application Serial Nos. 10/378,503 and 10/378,504, which
disclosures are incorporated herein by reference in their
entirety. The present application claims the benefit of
U.S. Provisional Application No.60/486,693, filed July 11,
10 2003, the entire disclosure of which is incorporated herein
by reference.

TECHNICAL FIELD

15 The present application relates to enterprise
management systems, and particularly, to a system and
method for assembling and deploying program components to a
cluster of servers in a network.

BACKGROUND

20 Middle tier applications, also referred to as
application servers, typically operate on top of a wide
range of existing enterprise systems such as database
management systems, transaction monitors, and naming and
directory services. Many of these application servers are
25 built based on standard specifications such as the Java 2
Platform, Enterprise Edition (J2EE) to provide portability
and scalability to applications managing and accessing
various enterprise systems.

J2EE, for example, defines a specification for

developing enterprise applications to follow as a standard. J2EE bases the enterprise applications on standardized, modular components, by providing a set of services to those components, and by handling many details of application behavior automatically. J2EE includes support for Enterprise JavaBeans (EJB) components, Java Servlets API, JavaServer Pages, and Extended Marked-up Language (XML) technology.

Accordingly, an application built conforming to the J2EE standard specification may be deployed to an application server, local or remote, or a cluster of application servers that supports the J2EE standards, thus allowing the deployed application to manage and access various resources provided by the underlying enterprise systems via the application server.

Clusters are sets of servers that are managed together and participate, for example, in workload management. The servers that are members of a cluster can be on different host machines. An example of an application server that supports clustering is the Websphere application server. Websphere is a set of Java-based tools from IBM that allows customers to create and manage business web sites with Java applications or servlets.

Briefly, deployment is the process of distributing and configuring various part of application programs such as J2EE applications to appropriate locations in application servers. In many cases, deployment process is vendor specific and requires manual steps. Further, deploying applications to a remote or cluster of application servers can require additional knowledge and steps.

SUMMARY

A system and method for automatically deploying program units to a cluster of application servers is provided. The method in one aspect includes assembling one or more program units for deploying to a cluster of networked servers. The method also includes retrieving information related to the cluster of networked servers, generating deployment descriptors from the information, and deploying the one or more program units to the cluster using at least the deployment descriptor.

In another aspect, the assembling and retrieving information may be performed interactively utilizing a user interface that retrieves information related to the one or more program units and user preferences from the user.

The system for automatically deploying program units to a cluster of networked servers includes data source management module operable to retrieve data source information from an application server to which to deploy one or more program units. The system also includes cluster management module operable to retrieve cluster information related to the application server, and container management module operable to retrieve container information related to the application server. The data source information, cluster information, and container information are used to automatically deploy the one or more program units to a cluster of networked servers.

Further features as well as the structure and operation of various embodiments are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers indicate identical or

functionally similar elements.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an architectural diagram illustrating the
5 cluster deployment flow in one embodiment.

Fig. 2 is a functional diagram illustrating an
overview of the cluster deployment flow in one embodiment.

DETAILED DESCRIPTION

10 The present disclosure describes a system and method
for assembling and deploying software or program units,
also referred to as program files or program modules to a
middle tier applications or application servers such as the
IBM Websphere cluster. EJBs to be deployed need not have
15 been generated by a particular assembly tool provided in
the target application server. The system and method in
one embodiment uses a cluster deployment flow (CDF)
architecture to allow users to automatically deploy program
units to an application server cluster.

20 The system and method, in one embodiment, may be
implemented as a plug-in, for example, to be used in
conjunction with a universal deployment tool such as the
one described in the co-pending U.S. patent application
serial no. 10/378,504. A user interface such as the one
25 described in that application may be provided to guide a
user through this single deployment process.

Assembling EJB components typically includes packaging
classes into JAR (Java archive) and WAR (Web archive)
files, creating EAR (EJB archive) file, and generating
30 deployment descriptor files, among other tasks. When
operated in conjunction with the deployment tool described

in the U.S. patent application serial no. 10/378,504 or a similar tool, the CDF system and method of the present disclosure in one embodiment allows assembling of one or more EJBs, generating application server suitable

5 deployment descriptors, and deploying the components to the application server cluster such as the Websphere cluster.

In the following description, well-known functions and components are not described in detail. Thus, many known features and definitions related to J2EE standard
10 specification and Java programming methods are not described in detail. Also in the following description, a Websphere cluster is used as an example of a target application server cluster, however, it should be understood that the disclosed method and system may be used
15 to deploy to other application servers having similar deployment schemes.

Fig. 1 is an architectural diagram illustrating the cluster deployment flow in one embodiment. An application server cluster 102 may include application servers on node
20 1 104, node 2 106, node 3 108, and node n 110 on supported platforms such as Windows, Linux, Solaris, AIX, OS/400 and/or OS/390. Clusters are sets of servers that are managed together and participate in workload management.

The cluster deployment flow (CDF) in one embodiment
25 includes application server (such as Websphere) administrative management components comprising data source management 114, cluster management 116, and container management 118. The data source management 114 retrieves and processes the information about the data sources that
30 are available for each node configured in the cluster. For instance, in the case of the Websphere application server,

data source management 114 retrieves data sources from a network deployment server of the application server 112. The retrieved data sources are presented to the user. The user may change the data source information or proceed with the existing data source bundled with the EJB.jar file.

An example of the retrieved data sources include shipment component data source such as "ShipmentComponent(cells/WebSphere2Network/nodes/WebSphere2:resources.xml#DataSource_1)." This information is filtered and retrieved from an existing application server's deployment server such as the Websphere 5.0 Network deployment server 112 using, for example, JACL script. Thus, in one embodiment, the data source management lists the data sources.

The cluster management 116 retrieves the cluster information and provides the cluster information to the user to select the appropriate target cluster to deploy the EJB. For instance, the cluster management 116 retrieves and processes the information of the cluster that are configured for an application server's deployment server such as the Websphere 5.0 Network deployment server. An example of cluster information is "snscluster1(cells/WebSphere2Network/clusters/snscluster1:cluster.xml#snscluster1)." Snscluster1 is a cluster in network deployment server. The information related to the snscluster1 is retrieved, for example, using a JACL script.

The container management 118 retrieves and processes information about the cell, node, and applications that are deployed on the cluster of application servers, virtual hosts for each node in a cluster for all the cluster that are configured for the application server's deployment

server. For example, the container management 118
retrieves the application names, nodes, server names, and
virtual hosts information using a JACL script. Examples of
retrieved information include

5 "WebSphere2(cells/WebSphere2Network/nodes/WebSphere2:node.xml#Node_1)," where WebSphere2 is the name of a node;
"server1(cells/WebSphere2Network/nodes/WebSphere2/servers/server1:server.xml#Server_1)," where server1 is a server;
default_host(cells/WebSphere2Network:virtualhosts.xml#VirtualHost_1)," where default_host is a virtual host; and
10 "ShipmentTechnologySamples," where shipment and technology samples are the applications already deployed onto the network deployment server.

Briefly, a cell is a repository of nodes in an
15 application server such as the Websphere Application Server. Cell is arbitrary, logical grouping of one or more nodes in an application server distributed network. A node refers to an application server installed on a machine. A node is within a cell in the application server. A cell
20 can have multiple nodes. In one embodiment, a cell is unique to a machine and a node is a logical grouping of managed servers.

A virtual host is a configuration enabling a single
host machine to resemble multiple host machines. Resources
25 associated with one virtual host do not share data with resources associated with another virtual host, even if the virtual hosts share the same physical machine. Virtual hosts allow the administrator to isolate, and independently manage, multiple sets of resources on the same physical
30 machine.

Servers that belong to a cluster are members of that

cluster set and all have identical application components deployed on them. Other than the applications configured to run on them, cluster members need not share any other configuration data. One cluster member may be running on a huge multi-processor enterprise server system while another member of that same cluster may be running on a small laptop. The server configuration settings for each of these two cluster members may be very different, except in the area of application components assigned to them. In that area of configuration, they are identical.

In one embodiment, the container management indicates the final stage of a deployment process, which creates all the appropriate JNDI bindings and deploys the EJB through the JACL script.

In one aspect, a user is provided with a choice to deploy new program modules or to redeploy existing program modules. If the user selected to deploy new application program modules, the program modules such as the Enterprise JavaBeans are deployed to the target application server for that cluster to a set of nodes. In this case, the node in a cluster is cloned and the program modules are deployed to the target node as well as the cloned node for that cluster. For example, referring to Fig. 1, the program modules are deployed to node 1 and cloned node 11 104. Clones may be created, for instance, by users during cluster creation in the Websphere. For example, during the cluster creation in the Websphere, a user can add a node and clone that specific node. Thus, the cluster will have a set of nodes in which the nodes in the set may be cloned by the user while creating the cluster. Creating clusters in the Websphere application servers are known to those

skilled in the art, therefore, will not be further described.

In one aspect, a script, for example, a JACL script is generated automatically to use the third party application
5 server's specific utility (such as wsadmin for Websphere) to deploy the program modules.

If the user selects an existing application from a cluster, then a script such as the JACL script (scripting language for Java platforms) may be generated to use the
10 third party application server's specific utility (such as wsadmin for Websphere) to re-deploy the program modules for that cluster to a set of nodes having the application servers installed.

Fig. 2 is a functional diagram illustrating an
15 overview of the cluster deployment flow in one embodiment. The enterprise archive (EAR) of the Enterprise JavaBeans, which may include an Enterprise JavaBean and Web application are generated and packaged from such tools as Joe, Gen, Plex, 2E, or other tools. At 204, the EAR are
20 accessed through the Universal Deployment tool (UDT) of Advantage Joe 3.1, for deployment to a deployment server such as the Websphere 5.0 network deployment server. At 206, the UDT dynamically loads the application server plugin such as the Websphere 5.0 Plugin to perform the
25 cluster deployment. At 208, specific application server profile such as the Websphere 5.0 Profile is created for the target Websphere Network deployment server.

At 210, information about the cluster is retrieved from a deployment server such as the network deployment
30 server for the Websphere applications (211). At 212, information about the nodes, servers, virtual hosts, and

applications deployed are retrieved from the deployment server such as the network deployment server for the Websphere applications (211). A cluster may have multiple nodes and clones.

5 A user has a choice to do the cluster deployment or regular deployment using the Single Transparent Deployment Flow (STDF). At 214, if cluster deployment is selected, then the cluster information, virtual hosts information, and applications already deployed onto clusters are listed.
10 At 216, a user may select cluster, virtual hosts, and existing applications for redeployment or new applications for new deployment. At 214, if regular node deployment using STDF is selected, regular node deployment is performed at 232, for example, as described in U.S. Patent
15 Application Serial No. 10/378,504.

A user may be given a choice to expose the Enterprise Java Beans as web services at 218. For instance, a user has a choice to select the enterprise application and deploy it as web services. The enterprise application, for
20 example, JavaBeans and points are generated for web services. A web service can be published in a directory where a web service browser can locate the information needed to access the web service. Since the enterprise Javabean (EJB) can be accessed as a web service which can
25 appear in a directory, the EJB is considered to be exposed. At 220, if a user selects to expose the Enterprise Java Beans as web services, then at 222, web services customization is done to expose the EJB as web services. The customization may include creating the interface
30 description information, additional classes and files needed to allow the EJB to be accessed as a web service.

If at 218, no EJBs are to be exposed as web services, deployment process takes place at 224.

Deployment process, for example, may include generating Apache SOAP deployment descriptor for the EJB
5 and executing the Websphere IBM "SoapEarEnabler.bat" command line utility. This tool may be used to package one or more web services in a Websphere EAR file. After executing this utility, one or more new WAR files will have been added to the EAR file. Deployment process may also
10 include generating a WSDL document for the EJB session bean using the Apache Axis Java2WSDL generation tool. The resulting WSDL document may then be added to the SOAP WAR file so that it is hosted by the web service web application.

15 At 226, a user may choose to customize any one or more of the deployment descriptor files such as ejb-jar.xml, web.xml, and application.xml files. For instance, a user may edit the ejb-jar.xml or web.xml or application.xml to include security tags or any websphere specific tags if
20 required. This customization involves a merge after the user edits the ejb-jar.xml or application.xml or web.xml file. The merge updates the changes made by the user to the appropriate ejb-jar.xml or application.xml or web.xml files and creates the EJB.jar and EJB.ear files.

25 At 228, EJB.ear is processed by creating the appropriate JNDI bindings, Websphere deployment descriptor files and deploys it to the target cluster of the network deployment server. JNDI bindings, for example, are created for each of the EJB.jar in the EJB.ear file. The JNDI
30 bindings have the information about the EJB JNDI name and the datasource information. The META-INF/ibm-ejb-jar-

bnd.xml META-INF/ibm-ejb-jar-ext.xml are created in each of the EJB.jar file and packaged into EJB.ear. The processing at 228 is performed, for example, when a user clicks a "finish button" on a user interface or otherwise indicates the user is finished with selecting available options and is ready for deployment.

At 230, the status of the deployment is reported back in the UDT. For instance, the deployment status is retrieved from the Websphere network deployment server.

10 The deployment status is retrieved from the deployment JACL scripts. Deployment status in one embodiment indicates whether the EJB deployed successfully or failed. The appropriate IBM Websphere exception message may be displayed onto a deployment console.

15 The system and method of the present invention may be implemented and run on a general-purpose computer. The embodiments described above are illustrative examples and it should not be construed that the present disclosure is limited to these particular embodiments. Various changes and modifications may be effected by one skilled in the art without departing from the spirit or scope of the invention as defined in the appended claims. For example, although the system and method disclosed herein has been described with respect to J2EE and Java environment, and Websphere application server as an example for ease of explanation, it is not limited only to such programming environment. Accordingly, the present invention is not limited except as by the appended claims.